

⑫ 公開特許公報 (A)

平1-287752

⑤Int.Cl.
G 06 F 11/28識別記号
A-7343-5B

⑬公開 平成1年(1989)11月20日

審査請求 未請求 請求項の数 1 (全7頁)

⑭発明の名称 デバッグ装置

⑪特 願 昭63-117640
⑫出 願 昭63(1988)5月13日

⑬発明者	工 藤 健 治	神奈川県川崎市中原区上小田中1015番地 内	富士通株式会社
⑬発明者	高 島 正 貴	神奈川県川崎市中原区上小田中1015番地 内	富士通株式会社
⑬発明者	赤 岩 升 一	神奈川県川崎市中原区上小田中1015番地 内	富士通株式会社
⑭出願人	富士通株式会社	神奈川県川崎市中原区上小田中1015番地	
⑭出願人	富士通マイコンシステムズ株式会社	神奈川県川崎市中原区上小田中1015番地	
⑭代理人	弁理士 井桁 貞一	外2名	

最終頁に続く

明細書

1. 発明の名称

デバッグ装置

オペレータによって任意のシンボル情報が入力される入力手段(6)からのシンボル情報に従ってシンボルテーブル(5)を検索し、

該シンボル情報に相関するアドレス情報を取り出して前記読み出し手段(2)に出力するアドレス情報出力手段(7)とを含むホスト側コンピュータ(8)を備えたことを特徴とするデバッグ装置。

2. 特許請求の範囲

多数の命令およびデータが各々アドレスを付与されて格納された記憶手段(1)と、

所定のアドレス情報に従って、該記憶手段(1)を検索し、アドレス情報で指定されたアドレス内の命令あるいはデータを読み出す読み出し手段(2)と、

読み出し手段(2)によって読み出された命令あるいはデータを実行する実行手段(3)とを含むターゲット側コンピュータ(4)を備えるとともに、

前記記憶手段(1)内の所定の命令あるいはデータに対応するシンボル情報およびこのシンボル情報に相関する前記記憶手段(1)のアドレスが書き込まれたシンボルテーブル(5)と、

3. 発明の詳細な説明

(概要)

デバッグ装置に関し、

シンボルテーブルをデバッグ対象のプログラム外に置くことにより、該プログラムに充分な機能を搭載しつつ、シンボリックデバッグのサポートを可能にしたデバッグ装置を提供することを目的とし、

多数の命令およびデータが各々アドレスを付与されて格納された記憶手段と、

所定のアドレス情報に従って、該記憶手段を検

索し、アドレス情報で指定されたアドレス内の命令あるいはデータを読み出す読み出し手段と、

読み出し手段によって読み出された命令あるいはデータを実行する実行手段とを含むターゲット側コンピュータを備えるとともに、

前記憶手段内の所定の命令あるいはデータに対応するシンボル情報およびこのシンボル情報に相関する前記憶手段のアドレスが書き込まれたシンボルテーブルと、

オペレータによって任意のシンボル情報が入力される入力手段からのシンボル情報に従ってシンボルテーブルを検索し、

該シンボル情報に相関するアドレス情報を取り出して前記読み出し手段に出力するアドレス情報出力手段とを含むホスト側コンピュータを備えて構成している。

〔産業上の利用分野〕

本発明は、デバッグ装置に関し、特に、ワンチップマイクロコンピュータ等のシングルボードコ

のサポートツールを使用して行われる。

〔従来の技術〕

従来のこの種のデバッガとしては、例えば同一のマイクロコンピュータ上にデバッグ対象のプログラムとデバッガを置き、デバッガを起動させてデバッグ作業を行うものがあり、具体的には、

- ① オペレータが、オブジェクトプログラムの絶対番地を意識しながらデバック作業を行う比較的に低レベルのものや、
- ② オペレータが、オブジェクトプログラムのモジュール等に付加された参照名（以下、シンボル情報）を意識しながらデバッガ作業を行う比較的に高レベルのもの（いわゆるシンボリックデバッガ）などがある。特に、シンボリックデバッガでは、デバッグ段階を通して頻繁に変更・修正がなされる絶対番地を意識しなくてもよいので、作業ミス等を防止することができ、デバッガ作業の効率が高い。

ンピュータに搭載されたプログラムに対して、充分な機能を搭載しつつ、シンボリックデバッガのサポートを可能にしたデバッガ装置に関する。

近年、半導体の微細化技術や製造技術の進展に伴って、例えば、ワンチップマイクロコンピュータは一段と高性能・低価格化してきた。これにより広範な産業分野でマイクロコンピュータが使用されるようになり、融通性、柔軟性に優れた多品種、少量のソフトウェアの開発比重が増大している。

一般に、ソフトウェアの開発段階では、ソースプログラムの作成作業およびソースプログラムから絶対番地指定のオブジェクトプログラムへの翻訳作業などの各種作業が行われるが、なかでもプログラムミスや修正のためのデバッガ作業は不可欠であり、特に、多品種、少量のソフトウェア開発ではデバッガ作業が頻繁に行われる。デバッガ作業は、任意の命令を実行させて、その結果を確認したり、あるいはデータ等を表示させたりするもので、通常、デバッガと呼ばれるソフトウェア

〔発明が解決しようとする課題〕

しかしながら、このような従来のシンボリックデバッガにあっては、デバッガ作業の効率が高いといった長所を有している反面、シンボル情報と絶対番地との相関表（以下、シンボルテーブル）を備える必要があり、一般にシンボルテーブルの容量は、オブジェクトプログラム容量の3倍程度を要することが多いから、メモリ容量の少ない、例えばワンチップマイクロコンピュータ等のシングルボードコンピュータにシンボリックデバッガを搭載した場合、メモリ容量に占めるデバッガの割合が増大することとなり、その結果、必要な本来の機能の搭載が制限されることがあった。このため、シングルボードコンピュータでは上記①の比較的低レベルのデバッガを適用せざるを得ず、デバッグ環境の面で問題点があった。しかも、シングルボードコンピュータは、一般に多品種、少量向けのものが多く、デバッグ環境の問題は大きい。

本発明は、このような問題点に鑑みてなされた

もので、シンボルテーブルをデバッグ対象のプログラム外に置くことにより、該プログラムに充分な機能を搭載しつつ、シンボリックデバッグのサポートを可能にしたデバッグ装置を提供することを目的としている。

(課題を解決するための手段)

第1図は本発明のデバッグ装置の原理プロック図を示す。

第1図において、多数の命令およびデータが各々アドレスを付与されて格納された記憶手段1と、所定のアドレス情報を従って、該記憶手段1を検索し、アドレス情報で指定されたアドレス内の命令あるいはデータを読み出す読出し手段2と、

読出し手段2によって読み出された命令あるいはデータを実行する実行手段3とを含むターゲット側コンピュータ4を備えるとともに、

前記記憶手段1内の所定の命令あるいはデータに対応するシンボル情報およびこのシンボル情報に相関する前記記憶手段1のアドレスが書き込み

れたシンボルテーブル5と、

オペレータによって任意のシンボル情報が入力される入力手段6からのシンボル情報に従ってシンボルテーブル5を検索し、

該シンボル情報に相関するアドレス情報を取り出して前記読出し手段2に出力するアドレス情報出力手段7とを含むホスト側コンピュータ8を備えて構成している。

(作用)

本発明では、デバッグ対象のプログラムを格納した記憶手段1を備えたターゲット側コンピュータ4の外に、シンボルテーブル5を備えたホスト側コンピュータ8が設けられ、シンボルテーブル5はオペレータによって入力されたシンボル情報で参照され、対応するアドレス情報が読み出される。そして、上記記憶手段1はこのアドレス情報によってアドレス指定され、該当するアドレス内の命令あるいはデータが取り出されて実行される。

したがって、オペレータはデバッグ作業の間、

頻繁に変更・修正されるアドレス情報（絶対番地で表現）を意識せずに、デバッグ作業の全段階を通して一定のシンボル情報に従って該作業を行うことができ、デバッグ環境の改善が図られる。

また、シンボルテーブル5は、デバッグ対象のプログラム外に置かれるので、記憶手段1の容量の大部分を該プログラムに割り当てることができ、必要な機能を充分に搭載することができる。

(実施例)

以下、本発明を図面に基づいて説明する。

第2、3図は本発明に係るデバッグ装置の一実施例を示す図である。

まず、構成を説明する。第2図において、10はターゲット側コンピュータとしてのワンチップマイクロコンピュータ（あるいはシングルボードコンピュータ）、20はホスト側コンピュータ、30はキーボードや表示装置あるいはプリンター等を備えた入力手段としての端末であり、これら、ワンチップマイクロコンピュータ10、ホスト側コンピ

ュータ20、端末30の間は同一規格のデータバス（例えばR S C 2 3 2 C）で接続されている（図中2重線で示す）。

上記ワンチップマイクロコンピュータ10は、ROM（記憶手段1）10a、RAM10b、CPU（読出し手段2、実行手段3）10cおよびI/O10dを含んで構成されている。ROM10aには、多数の命令およびデータが各々キャラクタ列で表現され、各キャラクタ列にアドレスが割り当てられたオブジェクトプログラムPGおよびマルチデバッガDBが格納されている。CPU10cは、オブジェクトプログラムPGを逐次実行して所定の演算処理を行うとともに、I/O10dを介して外部から所定のコマンド情報CMが取り込まれたときは、このCMに従ってマルチデバッガDBを実行させる。なお、オブジェクトプログラムPGおよびマルチデバッガDBはRAM10b上にあってもよい。

一方、上記ホスト側コンピュータ20は、ROM20a、ROM20b、RAM20c、I/O20d、I

ROM20eおよびCPU(出力手段7)20fを含んで構成されている。ROM20aには、ワンチップマイクロコンピュータ10とホスト側コンピュータ20間のデータ授受を制御する通信プログラムが格納されており、また、RAM20cには、必要に応じて外部の記憶装置等から転送されたシンボルテーブルTBLが格納される。なお、シンボルテーブルTBLは、予めROM20bに格納しておいてもよい。

ここで、シンボルテーブルTBLは、前記オブジェクトプログラムPG内の所定のキャラクタ列に対応するシンボル情報と、このシンボル情報に関連づけられた前記オブジェクトプログラムPGのアドレスとを相関表の形で表したもので、任意のシンボル情報を参照すると、当該シンボル情報に対応するキャラクタ列が格納されたオブジェクトプログラムPGのアドレスが取り出されるよう配列されたものである。

CPU20fは、オペレータによって端末30に入力されたシンボル情報を受け、シンボルテーブル

TBLを参照してアドレスを取り出した後、このアドレス情報を、通信プログラムに従ってワンチップマイクロコンピュータ10に送出する処理を実行する。また、ワンチップマイクロコンピュータ10から送られてきたデータを受け、このデータを端末30に転送して端末30に表示あるいはプリントアウトさせる処理も行う。

次に、作用を説明する。

今、オブジェクトプログラムPG内の所定のモジュールをダンプさせてモニターするデバッグ作業を考える。なお、上記所定のモジュールには、ソフトウェアの開発段階を通して一定の参照名(シンボル情報)が付与されており、ここではその参照名を便宜的に“!AAA”とする。

まず、オペレータは、端末30上のプロンプト表示“>”を確認した後、キーボードを操作して、ダンプ命令“D”を入力し、引き続いで参照名“!AAA”を入力する。ホストコンピュータ20のCPU20fは、“!AAA”に従って、シンボルテーブルTBLを参照し、“!AAA”に関連

づけられたアドレス(例えば“1000”)を読み出す。次いで、CPU20fは、ダンプ命令“D”および読み出された“1000”を組み合わせて、“D1000”なるコマンド文字列(コマンド情報CM)を作成する。そして、CPU20fは、通信プログラムを実行し、コマンド文字列をワンチップマイクロコンピュータ10に送出する。

ワンチップマイクロコンピュータ10のCPU10cは、送られてきたコマンド文字列を解読し、ダンプ命令であることを判別すると、オブジェクトプログラムPG内の“1000”番地から所定バイト数(例えば256バイト)分の命令やデータ等(以下、データ等)を読み出す。そして、CPU10cは読み出したデータ等をそのまま(キャラクタ列のまま)ホスト側コンピュータ20に転送する。ホスト側コンピュータ20のCPU20fはこの転送されたデータ等を端末30に送り、端末30上に表示させる。

以上のようなダンプ処理を概念的に表せば第3図の如く示される。第3図において、(SMBL

E)は多数のシンボル情報が書き込まれたシンボル領域、(ADRS)は上記(SMBLE)に関連づけてアドレス情報が書き込まれたアドレス領域、(キャラクタデータ)は命令およびデータ等である。

まず、端末30に入力された“!AAA”で(SMBLE)が参照され、同一シンボル情報のアドレス情報(“1000”)が取り出され、次いで、このアドレス情報“1000”で表示されたオブジェクトプログラムPGのアドレス内のキャラクタデータ(例えば、20、34、54、AF)が読み出されて端末30上に表示される。そして、アドレス“1000”から256バイトのダンプ処理であれば、上記キャラクタデータに引き続いで256バイトのデータが端末30に表示される。

このように本実施例では、オブジェクトプログラムPGとシンボルテーブルTBLとを分離し、ワンチップマイクロコンピュータ10の外の、ホスト側コンピュータ20にシンボルテーブルTBLを格納している。このため、オペレータはオブジェ

クトプログラムPGの絶対番地（例えば、上述の“1000”）を意識せずにデバッグ作業を行うことができる。すなわち、シンボル情報はソフトウエアの開発段階を通して一定であり、通常、変更・修正はなされないから、このシンボル情報を意識しながら行う本実施例のデバッグ作業は、作業の信頼性が高く効率が良い。

また、比較的に大きな容量となるシンボルテーブルTBLを、オブジェクトプログラムPGから切り離してホスト側コンピュータ20のRAM20c内に格納しているので、ワンチップマイクロコンピュータ10のROM10a内にオブジェクトプログラムPGを格納するに際し、これを余裕をもって行うことができる。すなわち、必要にして充分な機能をオブジェクトプログラムPGに持たせることができる。

さらに本実施例では、端末30、ホスト側コンピュータ20、ワンチップマイクロコンピュータ10の間を同一規格のデータバスで接続しているので、ホスト側コンピュータ20を介さずに、端末30とワ

ークに置いているので、ターゲット側コンピュータ内のプログラムに充分な機能を持たせることができる。

4. 図面の簡単な説明

第1図は本発明の原理ブロック図、

第2、3図は本発明の一実施例を示す図であり、

第2図はその全体構成図、

第3図はその作用を説明する概念図である。

- 1 ……記憶手段、
- 2 ……読み出し手段、
- 3 ……実行手段、
- 4 ……ターゲット側コンピュータ、
- 5 ……シンボルテーブル、
- 6 ……入力手段、
- 7 ……出力手段、
- 8 ……ホスト側コンピュータ、
- 10 ……ワンチップマイクロコンピュータ（ターゲット側コンピュータ）、

ンチップマイクロコンピュータ10と直接に接続することもできる。このようにすると、シンボリックデバッグは行えないものの、絶対番地指定のデバッグ作業は行える。例えば、端末30から“D1000”を入力すると、上記実施例と同一のダブル処理が実行される。したがって、ホスト側コンピュータ20を使用できない場合でも、従来通りのデバッグ作業を行うことはできる。

なお、ホスト側コンピュータ20とワンチップマイクロコンピュータ10の間にエミュレータを介在させる場合、このエミュレータに対応させてシンボルテーブルTBLを作成することにより、オペレータにエミュレータの存在を意識させないようにもできる。

〔発明の効果〕

本発明によれば、ターゲット側コンピュータ内のプログラムのデバッグに際し、シンボリックデバッグを行うことができる。

しかも、シンボルテーブルをホスト側コンピュ

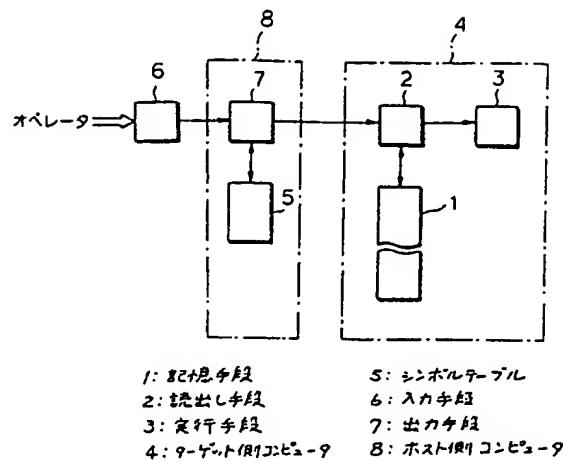
- 10a ……ROM（記憶手段）、
- 10c ……CPU（読み出し手段、実行手段）、
- 20 ……ホスト側コンピュータ、
- 20f ……CPU（出力手段）、
- TBL ……シンボルテーブル、
- 30 ……端末（入力手段）。

特許出願人 富士通株式会社

富士通マイコンシステムズ株式会社

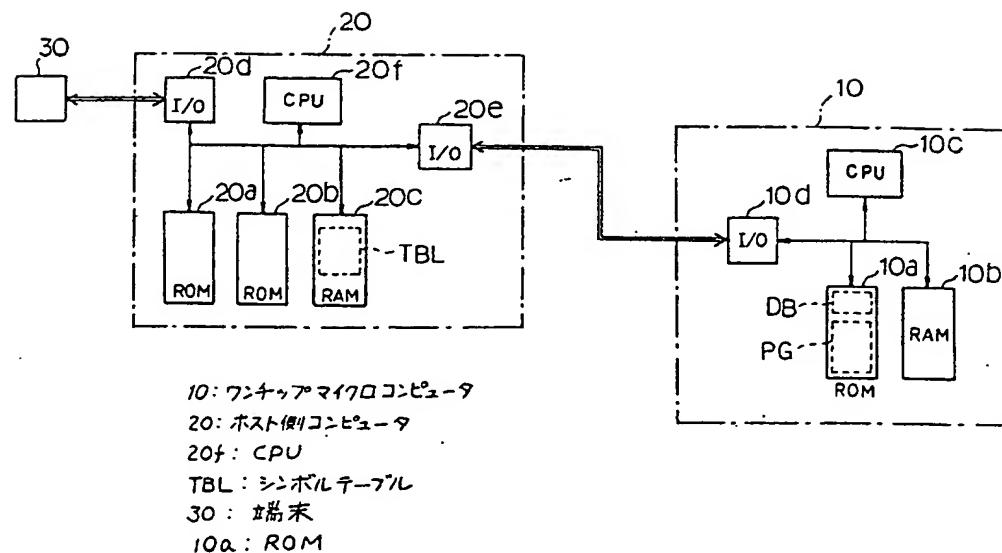
代理人 弁理士 井桁貞一





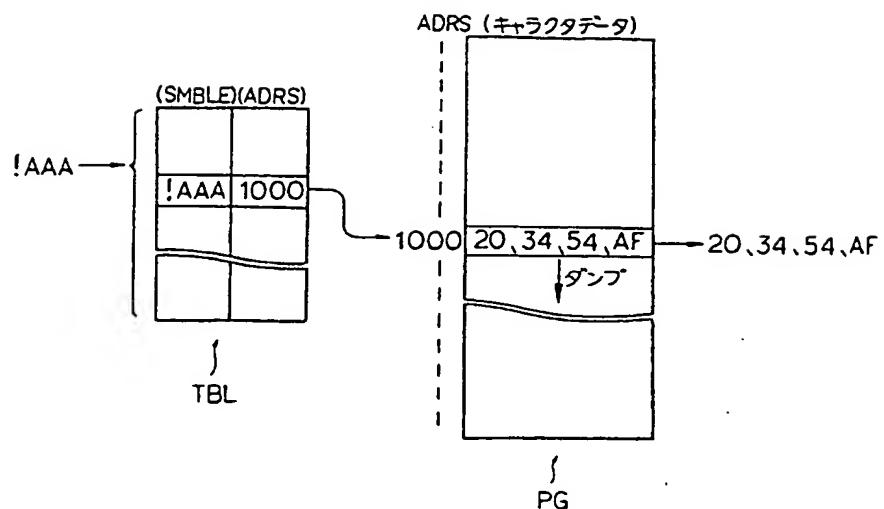
本発明の原理ブロック図

第1図



一実施例の全体構成図

第2図



一実施例の作用を説明する概念図

第3図

第1頁の続き

②発明者 阿部 信子 神奈川県川崎市中原区上小田中1015番地 内 富士通株式会社
 ②発明者 木村 芳孝 神奈川県川崎市中原区上小田中1015番地 富士通マイコン
 システムズ株式会社内

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 01-287752
(43)Date of publication of application : 20.11.1989

(51)Int.CI. G06F 11/28

(21)Application number : 63-117640

(71)Applicant : FUJITSU LTD
FUJITSU MICROCOMPUTER SYST LTD

(22)Date of filing : 13.05.1988

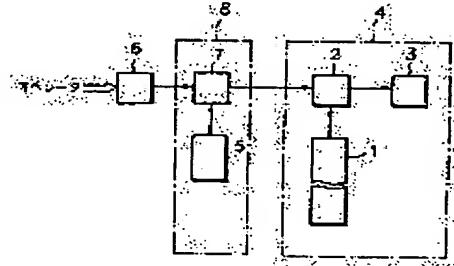
(72)Inventor : KUDO KENJI
TAKASHIMA MASAKI
AKAIWA SHOICHI
ABE NOBUKO
KIMURA YOSHITAKA

(54) DEBUGGING DEVICE

(57)Abstract:

PURPOSE: To improve debugging environment by retrieving a symbol table according to symbol information which is inputted by an operator, and fetching and outputting address information regarding the symbol information to a read-out means.

CONSTITUTION: A host-side computer 8 equipped with the symbol table 5 is provided in addition to a target-side computer 4 equipped with a storage means 1 stored with a program to be debugged, and the symbol table 5 is referred to by using the symbol information which is inputted by the operator to read out corresponding address information. Then the address of the storage means 1 is specified with this address information and an instruction or data is read out of the corresponding address and executed. Consequently, while the operator can perform the debugging operation according to the constant symbol information throughout the operation without considering address information which is changed or corrected frequently.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office